视图就是一个存在于数据库中的虚拟表。

视图本身没有数据，只是通过执行相应的select语句完成获得相应的数据。

**目录**

[创建视图：](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282279)

[删除视图：](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282280)

[修改视图：](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282281)

[视图缩减业务逻辑](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282282)

[视图的执行算法：](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282283)

[视图的更新：](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282284)

[不可更新的视图：](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282285)

[关于视图的可插入性：insert](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282286)

[With check option的用法：](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282287)

[WITH LOCAL/cascade CHECK OPTION的用法：](http://space.itpub.net/spacecp.php?action=spaceblogs&op=add#_Toc367282288)

创建视图**：**

CREATE [OR REPLACE] [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]

    VIEW view\_name [(column\_list)]

    AS select\_statement

[WITH [CASCADED | LOCAL] CHECK OPTION]

mysql> create view v\_emp as select empno,ename,job from emp;

Query OK, 0 rows affected (0.00 sec)

删除视图：

Drop view view\_name;

修改视图**：**

mysql> alter view v\_emp as select empno,ename,job,deptno from emp;

Query OK, 0 rows affected (0.01 sec)

修改视图结构，即修改所使用的字段的名称（可以隐含基表的字段名称）：

mysql> alter view v\_emp(v1,v2,v3,v4) as select empno,ename,job,deptno from emp;

Query OK, 0 rows affected (0.00 sec)

视图缩减业务逻辑

视图用来隐藏复杂的业务逻辑，从join连接查询产生一个view。先使用 视图完成一定的逻辑，再在视图的基础上完成另外的逻辑。

通常，视图完成的逻辑都是相对比较基础的逻辑。

视图的执行算法：

存在两种执行算法：

1、  Merge：合并的执行方式，每当执行的时候，先将我们视图的sql语句与外部查询视图的sql语句，混合在一起，最终执行；

2、  Temptable：临时表模式，每当查询的时候，将视图所使用的select语句生成一个结果的临时表，再在当前的临时表内进行查询。

指的是一个视图是在什么时候执行，依据哪些方式执行；

**对于MERGE**，会将引用视图的语句的文本与视图定义合并起来，使得视图定义的某一部分取代语句的对应部分。

**对于TEMPTABLE**，视图的结果将被置于临时表中，然后使用它执行语句。

**对于UNDEFINED**，MySQL将选择所要使用的算法。如果可能，它倾向于MERGE而不是TEMPTABLE，这是因为MERGE通常更有效，而且如果使用了临时表，视图是不可更新

当用户创建视图时，mysql默认使用一种undefine的处理算法，就是会自动在合并和临时表内进行选择。

注意：

1、  尽量使用视图完成读操作

2、  如果使用视图，则需要注意，对视图的修改，也是对基表的修改，会即时生效；

3、  删除视图时，不会销毁实体表内的数据

4、  如果大家做的是外部接口，一个数据库多个应用，针对每一个应用，采用不同的视图接口。

视图的更新：

mysql> select \* from dept;

+--------+-------+-----------+

| deptno | dname | loc       |

+--------+-------+-----------+

|     10 | bsc   | puyang    |

|     11 | bts   | xuchang   |

|     12 | 0521  | zhengzhou |

+--------+-------+-----------+

3 rows in set (0.01 sec)

mysql> create view v\_dept as select deptno,dname from dept;

Query OK, 0 rows affected (0.00 sec)

mysql> select \* from v\_dept;

+--------+-------+

| deptno | dname |

+--------+-------+

|     10 | bsc   |

|     11 | bts   |

|     12 | 0521  |

+--------+-------+

3 rows in set (0.23 sec)

mysql> update v\_dept set dname='MSC' where deptno=12;（更新视图）

Query OK, 1 row affected (0.03 sec)

Rows matched: 1  Changed: 1  Warnings: 0

mysql> select \* from v\_dept;

+--------+-------+

| deptno | dname |

+--------+-------+

|     10 | bsc   |

|     11 | bts   |

|     12 | MSC   |

+--------+-------+

3 rows in set (0.00 sec)

mysql> select \* from dept;（基表对应的数据也更新）

+--------+-------+-----------+

| deptno | dname | loc       |

+--------+-------+-----------+

|     10 | bsc   | puyang    |

|     11 | bts   | xuchang   |

|     12 | MSC   | zhengzhou |

+--------+-------+-----------+

3 rows in set (0.00 sec)

删除视图的数据：

mysql> delete from v\_dept where deptno=10;

Query OK, 1 row affected (0.00 sec)

mysql> select \* from v\_dept;

+--------+-------+

| deptno | dname |

+--------+-------+

|     11 | bts   |

|     12 | MSC   |

+--------+-------+

2 rows in set (0.00 sec)

mysql> select \* from dept;（基表对应的记录也被删除了）

+--------+-------+-----------+

| deptno | dname | loc       |

+--------+-------+-----------+

|     11 | bts   | xuchang   |

|     12 | MSC   | zhengzhou |

+--------+-------+-----------+

2 rows in set (0.00 sec)

不可更新的视图：

某些视图是可更新的。也就是说，可以在诸如UPDATE、DELETE或INSERT等语句中使用它们，以更新基表的内容。对于可更新的视图，在视图中的行和基表中的行之间必须具有一对一的关系。还有一些特定的其他结构，这类结构**会使得视图不可更新**。更具体地讲，如果视图包含下述结构中的任何一种，那么它就是不可更新的：

·         聚合函数（SUM(), MIN(), MAX(), COUNT()等）。

·         DISTINCT

·         GROUP BY

·         HAVING

·         UNION或UNION ALL

·         位于选择列表中的子查询

·         Join

·         FROM子句中的不可更新视图

·         WHERE子句中的子查询，引用FROM子句中的表。

·         仅引用文字值（在该情况下，没有要更新的基本表）。

·         ALGORITHM = TEMPTABLE（使用临时表总会使视图成为不可更新的）。

**创建不可更新的视图**：（使用临时表的算法）

mysql> create algorithm=temptable view v\_dept2 as select \* from dept;

Query OK, 0 rows affected (0.00 sec)

mysql> show create view v\_dept2\G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                View: v\_dept2

         Create View: CREATE ALGORITHM=TEMPTABLE DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `v\_dept2` AS select `dept`.`deptno` AS `deptno`,`dept`.`dname` AS `dname`,`dept`.`loc` AS `loc` from `dept`

character\_set\_client: gbk

collation\_connection: gbk\_chinese\_ci

1 row in set (0.00 sec)

尝试更新视图，报错，（一定程度上保证了基表数据的安全性）

mysql> update v\_dept2 set loc='shanghai' where deptno=10;

ERROR 1288 (HY000): The target table v\_dept2 of the UPDATE is not updatable

关于视图的可插入性：insert

如果视图满足关于视图列的下述额外要求，**可更新的**视图也是可插入的：

·         不得有重复的视图列名称。

·         视图必须包含没有默认值的基表中的所有列。

·         **视图列必须是简单的列引用而不是导出列**。导出列不是简单的列引用，而是从表达式导出的。下面给出了一些导出列示例：

·                3.14159

·                col1 + 3

·                UPPER(col2)

·                col3 / col4

·                (subquery)

混合了简单列引用和导出列的视图是不可插入的，但是，如果仅更新非导出列，视图是可更新的。

**更改视图的算法**：（原来是temptable，改为merge，从而视图变成可以更新了）

mysql> alter ALGORITHM=merge view v\_dept2 as select \* from dept;

Query OK, 0 rows affected (0.00 sec)

mysql> select \* from v\_Dept2;

+--------+-------+-----------+--------+

| deptno | dname | loc       | amount |

+--------+-------+-----------+--------+

|     11 | bts   | xuchang   |      0 |

|     12 | MSC   | zhengzhou |      0 |

+--------+-------+-----------+--------+

2 rows in set (0.00 sec)

**由于该视图没有导出列，故可以插入数据**：

mysql> insert into v\_dept2 values(13,'SW','shanghai',100);

Query OK, 1 row affected (0.00 sec)

mysql> select \* from dept;（基表数据也被插入）

+--------+-------+-----------+--------+

| deptno | dname | loc       | amount |

+--------+-------+-----------+--------+

|     11 | bts   | xuchang   |      0 |

|     12 | MSC   | zhengzhou |      0 |

|     13 | SW    | shanghai  |    100 |

+--------+-------+-----------+--------+

3 rows in set (0.02 sec)

mysql> select \* from v\_dept2;（视图数据插入）

+--------+-------+-----------+--------+

| deptno | dname | loc       | amount |

+--------+-------+-----------+--------+

|     11 | bts   | xuchang   |      0 |

|     12 | MSC   | zhengzhou |      0 |

|     13 | SW    | shanghai  |    100 |

+--------+-------+-----------+--------+

3 rows in set (0.00 sec)

**更改视图含有导出列，则通过视图不能插入数据**：

mysql> alter view v\_dept2 as select \*,amount/2 from dept;（amount/2是导出列）

Query OK, 0 rows affected (0.01 sec)

mysql> select \* from v\_dept2;

+--------+-------+-----------+--------+----------+

| deptno | dname | loc       | amount | amount/2 |

+--------+-------+-----------+--------+----------+

|     11 | bts   | xuchang   |     90 |  45.0000 |

|     12 | MSC   | zhengzhou |     80 |  40.0000 |

|     13 | SW    | shanghai  |    100 |  50.0000 |

+--------+-------+-----------+--------+----------+

3 rows in set (0.02 sec)

**插入一条数据是不允许的：**

mysql> insert into v\_dept2(deptno,dname,loc,amount) values(14,'HW','puyang',110);

ERROR 1471 (HY000): The target table v\_dept2 of the INSERT is not insertable-into

**但是如果简单更新非导出列是可以的：**

mysql> update v\_dept2 set amount=110 where deptno=13;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1  Changed: 1  Warnings: 0

mysql> select \* from dept;（基表数据更新成功）

+--------+-------+-----------+--------+

| deptno | dname | loc       | amount |

+--------+-------+-----------+--------+

|     11 | bts   | xuchang   |     90 |

|     12 | MSC   | zhengzhou |     80 |

|     13 | SW    | shanghai  |    110 |

+--------+-------+-----------+--------+

3 rows in set (0.01 sec)

mysql> select \* from v\_dept2;（视图数据更新成功）

+--------+-------+-----------+--------+----------+

| deptno | dname | loc       | amount | amount/2 |

+--------+-------+-----------+--------+----------+

|     11 | bts   | xuchang   |     90 |  45.0000 |

|     12 | MSC   | zhengzhou |     80 |  40.0000 |

|     13 | SW    | shanghai  |    110 |  55.0000 |

+--------+-------+-----------+--------+----------+

3 rows in set (0.00 sec)

With check option的用法：

**（with check option对于没有where条件的视图不起作用的）**

mysql> alter view v\_dept2 as select \* from dept where dname='SW' with check option;

Query OK, 0 rows affected (0.00 sec)（只限于插入dname是SW的记录）

mysql> insert into v\_dept2 values(15,'SW','beijing',20);（插入成功）

Query OK, 1 row affected (0.00 sec)

mysql> select \* from dept;

+--------+-------+-----------+--------+

| deptno | dname | loc       | amount |

+--------+-------+-----------+--------+

|     11 | bts   | xuchang   |     90 |

|     12 | MSC   | zhengzhou |     80 |

|     13 | SW    | shanghai  |     60 |

|     14 | SW    | nanjing   |     65 |

|     15 | SW    | beijing   |     20 |

+--------+-------+-----------+--------+

5 rows in set (0.00 sec)

mysql> insert into v\_dept2 values(15,'BSC','nanjing',65);（插入失败）

ERROR 1369 (HY000): CHECK OPTION failed 'temp.v\_dept2'

mysql> delete from v\_dept2 where deptno=15;（有没有with check option，不影响删除操作）

Query OK, 1 row affected (0.00 sec)

mysql> update v\_dept2 set dname='HW' where deptno=13;（更新成非SW的 失败）

ERROR 1369 (HY000): CHECK OPTION failed 'temp.v\_dept2'

**对于with check option用法，总结如下**：

通过有with check option选项的视图操作基表(只是面对单表，对连接多表的视图正在寻找答案)，有以下结论： 插入后的数据，通过视图能够查询出来就符合WITH CHECK OPTION 否则就不符合；

首先视图只操作它可以查询出来的数据，对于它查询不出的数据，即使基表有，也不可以通过视图来操作。

1.对于update,有with check option，要保证update后，数据要被视图查询出来

2.对于delete,有无with check option都一样

4.对于insert,有with check option，要保证insert后，数据要被视图查询出来

对于没有where 子句的视图，使用with check option是多余的

WITH LOCAL/cascade CHECK OPTION的用法：

在关于可更新视图的WITH CHECK OPTION子句中，当视图是根据另一个视图定义的时，LOCAL和CASCADED关键字决定了检查测试的范围。LOCAL关键字对CHECK OPTION进行了限制，使其仅作用在定义的视图上，CASCADED会对将进行评估的基表进行检查。如果未给定任一关键字，默认值为CASCADED。

mysql> select \* from test;

+----+------+

| id | aa   |

+----+------+

|  1 |   12 |

|  2 |    4 |

|  3 |   44 |

|  4 |   25 |

|  5 |   26 |

|  6 |    8 |

|  7 |   15 |

+----+------+

7 rows in set (0.00 sec)

mysql> create view v1 as select \* from test where aa<40 with check option;（视图v1）

Query OK, 0 rows affected (0.02 sec)

mysql> select \* from v1;

+----+------+

| id | aa   |

+----+------+

|  1 |   12 |

|  2 |    4 |

|  4 |   25 |

|  5 |   26 |

|  6 |    8 |

|  7 |   15 |

+----+------+

6 rows in set (0.00 sec)

mysql> create view v2 as select \* from v1 where aa>10 with local check option;

Query OK, 0 rows affected (0.00 sec)

mysql> create view v3 as select \* from v1 where aa>10 with cascaded check option;

Query OK, 0 rows affected (0.00 sec)

mysql> insert into v2 values(null,50);（通过视图v2插入50，只检查插入的数据是否满足v2

的条件aa>10,成功插入）

Query OK, 1 row affected (0.00 sec)

mysql> insert into v3 values(null,50);（通过视图v3插入50，不仅检查是否满足V3的条件aa>10，还要检查是否满足v1的条件aa<40）插入失败

ERROR 1369 (HY000): CHECK OPTION failed 'temp.v3'

但是，虽然通过视图v2插入成功，v2中并没有50这条数据，test基表中有50这条数据，直接插入到基表了

mysql> select \* from v2;

+----+------+

| id | aa   |

+----+------+

|  1 |   12 |

|  4 |   25 |

|  5 |   26 |

|  7 |   15 |

+----+------+

4 rows in set (0.00 sec)

mysql> select \* from test;

+----+------+

| id | aa   |

+----+------+

|  1 |   12 |

|  2 |    4 |

|  3 |   44 |

|  4 |   25 |

|  5 |   26 |

|  6 |    8 |

|  7 |   15 |

|  8 |   50 |

+----+------+

8 rows in set (0.00 sec)